Limits to the Use of the Zachman Framework in Developing and Evolving Architectures for Complex Systems of Systems

Philip Boxer, Suzanne Garcia

May 6th 2009

Agenda

➢ The Enterprise/Architecture relationship

The demands of collaborative systems of systems

Limits to the use of the Zachman Framework & the consequences for DODAF 2.0

Summary

<table>
<tr><td colspan="3" align="center">Report Documentation Page</td><td>Form Approved
OMB No. 0704-0188</td></tr>
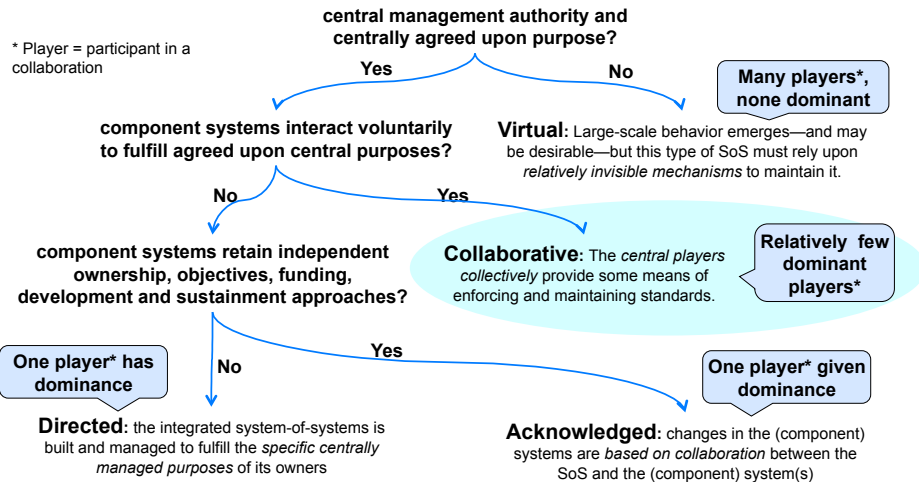</table>

Report Documentation Page			*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **06 MAY 2009**	2. REPORT TYPE	3. DATES COVERED **00-00-2009 to 00-00-2009**

4. TITLE AND SUBTITLE **Limits to the Use of the Zachman Framework in Developing and Evolving Architectures for Complex Systems of Systems**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Carnegie Mellon University,Software Engineering Institute,Pittsburgh,PA,15213**	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
presented at Software Engineering Institute (SEI) Architecture Technology User Network (SATURN) Conference, 4-7 May 2009, Pittsburgh, PA.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**11**	

Systems of Systems: *4 Types defined by OSD SE Guide for Systems of Systems*

* Player = participant in a collaboration

central management authority and centrally agreed upon purpose?

Yes   No

Many players*, none dominant

component systems interact voluntarily to fulfill agreed upon central purposes?

Virtual: Large-scale behavior emerges—and may be desirable—but this type of SoS must rely upon *relatively invisible mechanisms* to maintain it.

No   Yes

component systems retain independent ownership, objectives, funding, development and sustainment approaches?

Collaborative: The *central players collectively* provide some means of enforcing and maintaining standards.

Relatively few dominant players*

One player* has dominance

No   Yes

One player* given dominance

Directed: the integrated system-of-systems is built and managed to fulfill the *specific centrally managed purposes* of its owners

Acknowledged: changes in the (component) systems are *based on collaboration* between the SoS and the (component) system(s)

Source of definitions: Systems Engineering Guide for Systems of Systems, OSD, Version 1.0 August 2008. Brackets added.

Architectural Genres:
different genres for different purposes

The primary interfaces across genres as evidenced by working group discussions:

The enterprise is supported by an *infrastructure*

Enterprise Architecture

Enabler   *Quality attributes*

System of systems architecture   *Mutually constraining*   **System architecture**

Directed or Acknowledged Systems of Systems

Software architect mostly on the receiving end

Software architecture
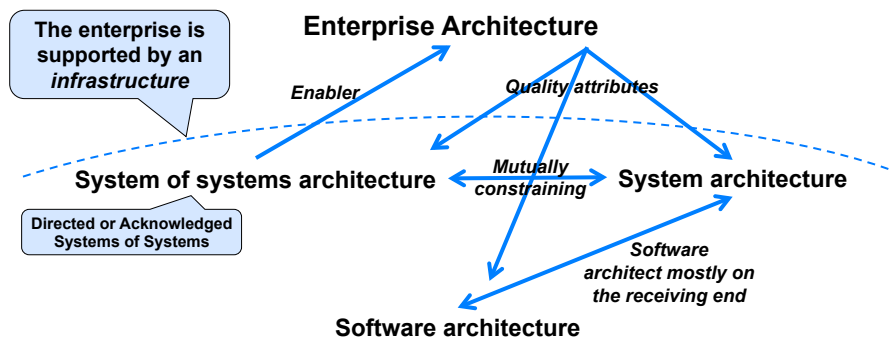
These genres reflect a *supply-side perspective* on the enterprise

Source: U.S. Army Workshop on Exploring Enterprise, System of Systems, System, and Software Architectures, John Bergey, Stephen Blanchette, Jr., Paul Clements, Mike Gagliardi, John Klein, Rob Wojcik, Bill Wood, March 2009 TECHNICAL REPORT CMU/SEI-2009-TR-008

The Enterprise Architecture defines the way it creates value: *Zachman roots to DODAF*

	DATA (WHAT) e.g. data	FUNCTION (HOW) e.g. function	NETWORK (WHERE) e.g. network	PEOPLE (WHO) e.g. organisation	TIME (WHEN) e.g. schedule	MOTIVATION (WHY) e.g. strategy
SCOPE (*Competitive* context) Planning	List of Things Important to the Business / Entity = Class of	List of Processes the Business Performs / Process = Class of	List of Locations in Which the Business Operates / Node = Major Business	List of Organizations Important to the Business / People = Major	List of Events/Cycles Significant to the Business / Time = Major Business	Lists of Business Goals/Strategies / Ends/Means = Major Business Goal/Strategy
BUSINESS MODEL (Conceptual) Owning	e.g. Semantic Model / Entity = Business Entity Relationship = Business Relationship	e.g. Business Process Model / Process = Business Process I/O = Business Resources	e.g. Business Logistics System / Node = Business Location Link = Business Linkage	e.g. Work Flow Model / People = Organization Unit Work = Work Product	e.g. Master Schedule / Time = Business Event Cycle = Business Cycle	e.g. Business Plan / End = Business Objective Means = Business Strategy
SYSTEM MODEL (Logical) Designing	e.g. Logical Data Model / Entity = Data Entity Relationship = Data Relationship	e.g. Application Architecture / Process = Application Function I/O = User Views	e.g. Distributed System Architecture / Node = I/S Function (Processor, Storage, etc.) Link = Line Characteristics	e.g. Human Interface Architecture / People = Role Work = Deliverable	e.g. Processing Structure / Time = System Event Cycle = Processing Cycle	e.g. Business Rule Model / End = Structural Assertion Means = Action Assertion
TECHNOLOGY MODEL (Physical) Building	e.g. Physical Data Model / Entity = Segment/Table/etc. Relationship = Pointer/Key/etc.	e.g. System Design / Process = Computer Function I/O = Data Elements/Sets	e.g. Technology Architecture / Node = Hdw/System Software Link = Line Specifications	e.g. Presentation Architecture / People = User Work = Screen Formats	e.g. Control Structure / Time = Execute Cycle = Component Cycle	e.g. Rule Design / End = Condition Means = Action
DETAILED REPRESENTATIONS (out-of-*modelling*-context) Subcontracting	e.g. Data Definition / Entity = Field Relationship = Address	e.g. Program / Process = Language Statement I/O = Control Block	e.g. Network Architecture / = Address / = Address	e.g. Security Architecture / People = Identity Work = Job	e.g. Timing Definition / Time = Interrupt Cycle = Machine Cycle	e.g. Rule Specification / End = Sub-condition

Source of coloured squares: Zachman Framework, www.zifa.com

Focus on defined value-creating relationships

The context defining that focus is the Enterprise

Agenda

The Enterprise/Architecture relationship
➢ The demands of collaborative systems of systems
Limits to the use of the Zachman Framework & the consequences for DODAF 2.0
Summary

Speech by Secretary Gates:
There are two paradigms that must coexist

The need for state of the art systems – particularly longer range capabilities – will never go away…

We also need specialized, often relatively low-tech equipment for stability and counter-insurgency missions.

- How do we institutionalize rapid procurement and fielding of such capabilities?
- Why do we currently have to go outside the normal bureaucratic process?

Our conventional modernization programs seek a 99% solution in years.
Stability and counter-insurgency missions require 75% solutions in months.

- The challenge is whether in our bureaucracy and in our minds these two different paradigms can be made to coexist.

Extracted from speech delivered by Secretary of Defense Robert M. Gates,
National Defense University, Washington, D.C. September 29, 2008 http://www.defenselink.mil/speeches/
speech.aspx?speechid=1279

The three tempos: *analyzing the impact of the enterprise's relation to customers' changing demands*
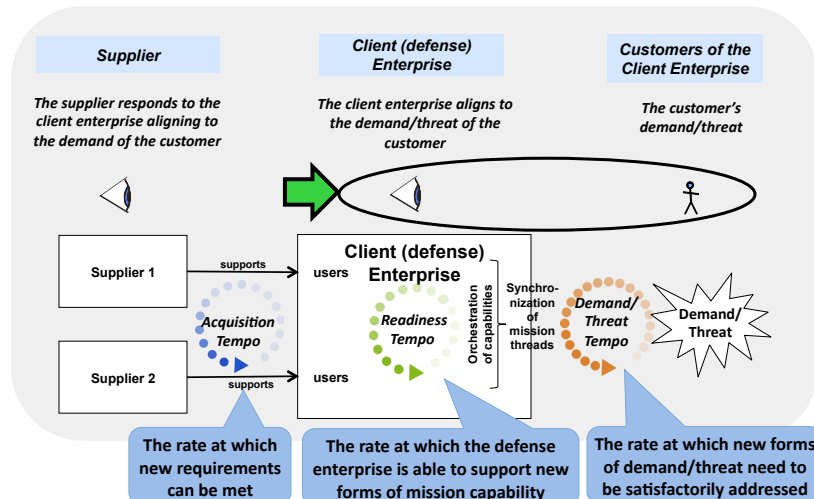
Managing diverging tempos: *the readiness tempo has to be managed in its own right*

The two paradigms are about diverging acquisition and demand/threat tempos
- Their coexistence depends on managing the readiness tempo in its own right

Managing the readiness tempo means:
- sustaining *multiple collaborations* between players able to address *concurrent* types of demand/threat
- building *organizational agility* into the supporting socio-technical infrastructures

---

Governance of a Collaborative SoS: *involves multiple collaborations with a supporting infrastructure*

The players in a collaboration can be spread across multiple enterprises and/or different parts of a single enterprise

↓ **Larger stakeholder context**

Collaborations of Players

Supporting Infrastructure

Multiple value-creating relationships

Governance
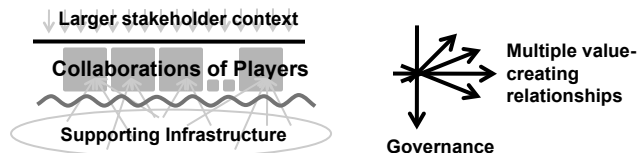
It is the players participating in a particular collaboration who will define
- Their system-of-interest and its environment
- The stakeholders they judge to be relevant
- The way they want their collaboration supported by the infrastructure

And so… *a demand-side perspective needs to be added*

Collaborative SoS present a different order of complexity

This complexity arises because
- multiple collaborations between players exist concurrently,
- each with its own relationship to demand/threat, and
- supported by a shared infrastructure

It means adding a demand-side perspective on the collaborations
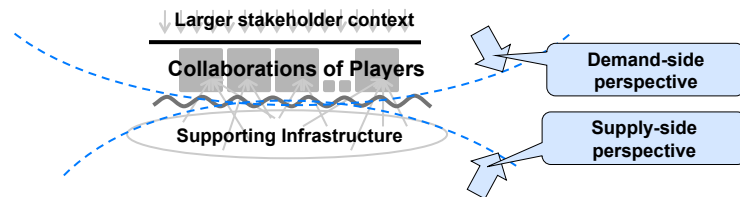
↓ Larger stakeholder context

Collaborations of Players

Supporting Infrastructure

Demand-side perspective

Supply-side perspective

Managing both paradigms: means managing the relationship between the two 'V's

Effects on Demand/Threat

… constrains what is possible up here

Scenarios

Demand/ Threat Tempo

Mission Command

Multiple Concurrent Collaborations

Force Structures

Composite Capabilities

demand-side
supply-side

Capability gap

Readiness Tempo

Operational Capability

Requirement

Solution

Design decomposition

Acquisition Tempo

System integration

System components

What happens down here…

Boxer, P.J. (2007) Managing the SoS Value Cycle, January 2007, http://www.asymmetricdesign.com/archives/85

Agenda

The Enterprise/Architecture relationship

The demands of collaborative systems of systems

➤ Limits to the use of the Zachman Framework & the consequences for DODAF 2.0

Summary

The demand-side perspective: *creates gaps in Zachman*

Source of gaps: Philip Boxer, *Modeling structure-determining processes, http://www.asymmetricdesign.com/archives/59, December 2006*

DODAF 2.0 Entities and Views:
what gets modeled?

Modeling Elements

DODAF TAXONOMY TYPES and CADM 2.0 PRINCIPAL INDEPENDENT ENTITIES	All Views (AV)		Operational View (OV)							System View (SV)											Tech View(TV)	
	1	(2)	1	(2)	(3)	(4)	(5)	(6)	(7)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(1)	(2)
Operational Nodes *Organizations, Types of Organizations, and Operational Roles*	▤	˝	▤	˝	˝	˝	˝	˝														
Performance Attributes	▤	˝	▤		ž		ž	ž					ž		˝	˝	ž	ž	ž		ž	ž
Technical Standards *Info Processing, Info Transfer, Data, Security, and Human Factors*	▤	˝	▤						ž	ž	ž		ž		ž		ž	ž		ž	˝	˝
Technology Areas	▤	˝	▤															˝				ž
Physical Nodes *Facilities, Platforms, Units, and Locations (including Features)*	▤	˝	▤							ž	ž	ž		ž	ž	ž	ž				ž	ž
Triggers/Events	▤	˝	▤		˝			˝								˝			˝			
Operational Activities *(and Tasks)*	▤	˝	▤	ž	˝			˝						˝								
Technology Areas	▤	˝	▤															˝				ž
Systems *Families-of-Systems, Systems-of-Systems, Networks, Applications, Software, and Equipment*	▤	˝	▤							˝	˝	˝	˝	ž	˝	˝	˝	˝			ž	ž
Information Elements *(and Data Elements)*	▤	˝	▤	ž				˝			˝		˝						ž	˝	ž	ž
System Functions	▤	˝	▤								˝		˝	ž	ž	ž	ž	˝			ž	ž

˝ = Taxonomy element plays a primary role   ž = element plays a secondary role   ▤ = unstructured text or graphics

Source: Fig 3-2, DoD Architectural Framework version 2.0 Volume III: Architecture Data Description, DOD Architecture Framework Working Group, July 2006
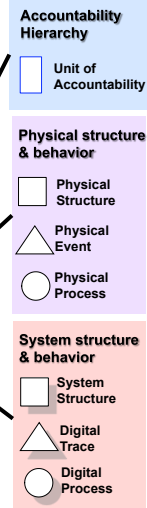
Modeling Elements

Accountability Hierarchy
- Unit of Accountability

Physical structure & behavior
- Physical Structure
- Physical Event
- Physical Process

System structure & behavior
- System Structure
- Digital Trace
- Digital Process

Entities not modeled by DODAF 2.0:
the demand-side perspective is not included
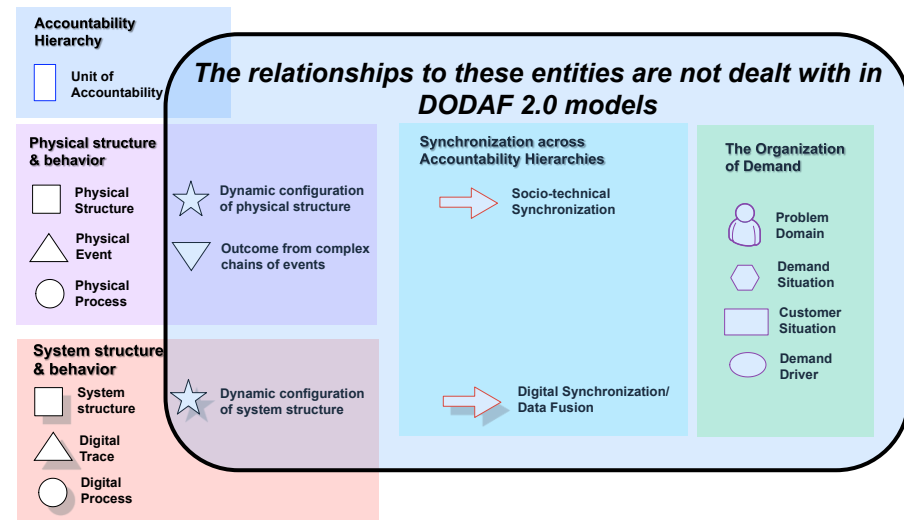
Accountability Hierarchy
- Unit of Accountability

Physical structure & behavior
- Physical Structure
- Physical Event
- Physical Process

System structure & behavior
- System structure
- Digital Trace
- Digital Process

The relationships to these entities are not dealt with in DODAF 2.0 models

- Dynamic configuration of physical structure
- Outcome from complex chains of events
- Dynamic configuration of system structure

Synchronization across Accountability Hierarchies
- Socio-technical Synchronization
- Digital Synchronization/ Data Fusion

The Organization of Demand
- Problem Domain
- Demand Situation
- Customer Situation
- Demand Driver

Describing the demand-side: *bridging the gaps*

Source of gaps: Philip Boxer, Modeling structure-determining processes. http://www.asymmetricdesign.com/archives/59, December 2006
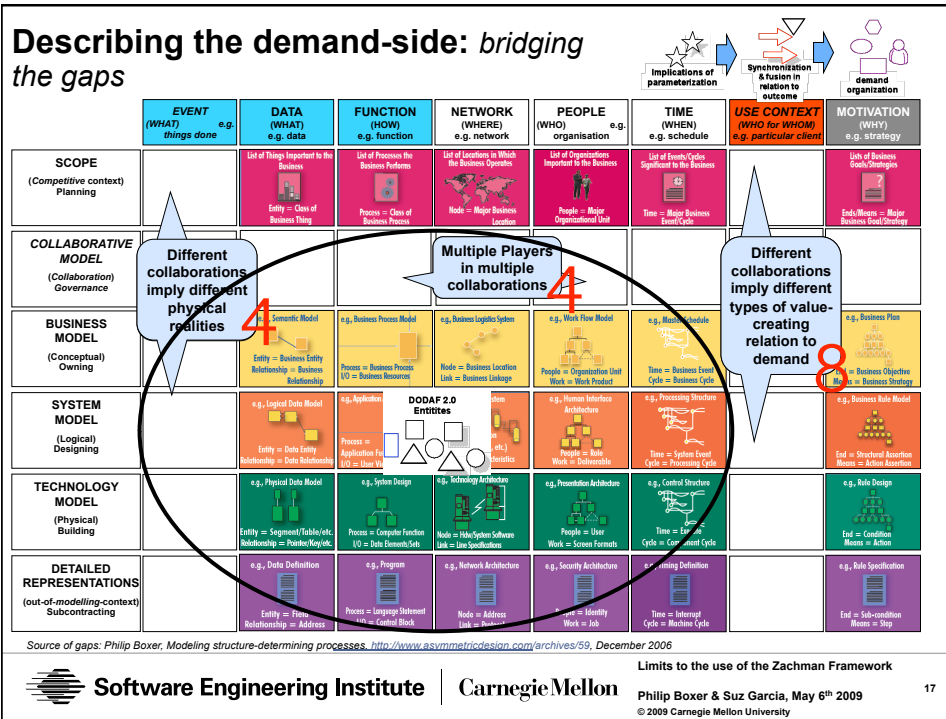
Agenda

The Enterprise/Architecture relationship

The demands of collaborative systems of systems

Limits to the use of the Zachman Framework & the consequences for DODAF 2.0

➹ Summary

Summary: *both supply-side and demand-side perspectives need to be modeled*

Supporting the development of collaborative systems of systems involves modeling more than the supply-side entities in Zachman-rooted representations like DODAF 2.0

- Including a demand-side perspective means being able to account for
  - cross-cutting synchronization, not just hierarchical accountability
  - multi-enterprise development and co-evolution
  - inherent variation in the way user's demands emerge and evolve
  - the resultant tempo of the ongoing development of systems of systems
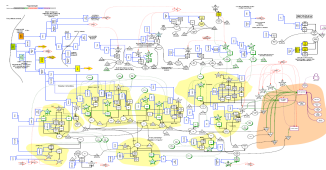
If you're a software architect…so what?

If you think/know you're involved in a SoS collaboration,

- It is likely that the requirements you are working to do NOT account for sufficient demand-side variety
  - Don't over-constrain your software architecture too early
  - Look for architectural mechanisms that can accommodate later information on interfaces and implementations
- Try to find out the level of awareness of SoS issues that is present on the part of your systems engineers
  - The more they are aware of their lack of control over organizational and technical interactions across the collaboration, the less likely they will be to pass down over-constraining architecture requirements to the software
  - If awareness of SoS issues is low, find out how they are planning to deal with some of the demand-side constructs discussed here
- Start thinking about your customers' "operations architecture" – the components and interfaces that they are operating with and that you are supporting with your software
  - Look for points of complementarity and conflict between your software architecture and your customer's "operations architecture"

Software Engineering Institute | **Carnegie Mellon**

Limits to the use of the Zachman Framework
Philip Boxer & Suz Garcia, May 6th 2009
© 2009 Carnegie Mellon University

21

Contact Information

Philip Boxer, Suzanne Garcia

Research, Technology and Systems Solutions Program,
Software Engineering Institute, Carnegie Mellon University

Email: **pboxer@sei.cmu.edu**, **smg@sei.cmu.edu**

Mail:
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Software Engineering Institute | **Carnegie Mellon**

Limits to the use of the Zachman Framework
Philip Boxer & Suz Garcia, May 6th 2009
© 2009 Carnegie Mellon University

22